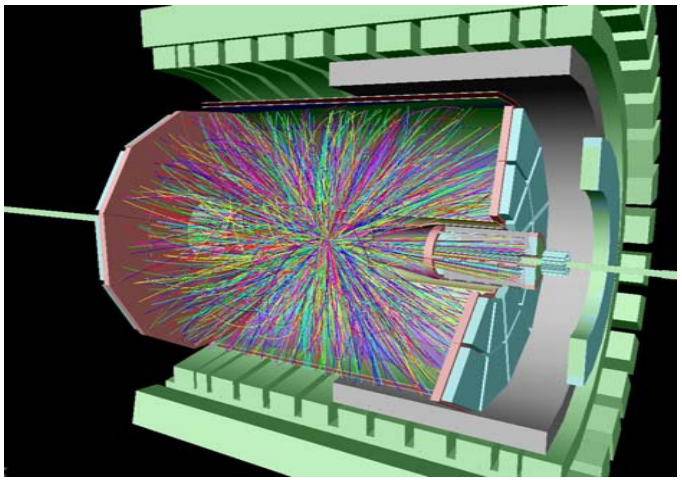
Qt/QtRoot in STAR and EventDisplay development



http://www.star.bnl.gov/STAR/comp/vis/

Valeri Fine (BNL, STAR)

STAR

BROOKHAVEN NATIONAL LABORATORY

Towards affordable GUI

Create pre-conditions, hardware and software infrastructure making the GUI in affordable for “everyday” STAR applications

Qt framework + Qt GUI

+

ROOT Data-Analysis and Visualization facility

=

New generation of the powerful interactive applications for STAR

Whether it is feasible to benefit just from the two best technologies

Qt-layer for ROOT (ROOT layer for Qt) - the unique technology creating the complex data-analysis and the computer simulation interactive software packages and problem-oriented applications based on Qt and ROOT frameworks.

The system has been under development for a few years within STAR and it is included in the regular distribution of the ROOT package. The later is a “de facto” standard of the HENP OO frameworks for the modern HENP experiments

Batch and interactive stages

- GUI is not the first concern of the HENP frameworks. We worry whether we are capable to collect, preserve, re-distribute our hard-earned TBytes.
- However the final stages of the job are mainly interactive.
- The very first steps of the data-taking in the “control rooms” are interactive also

Operating system “model” for HENP cross-platform applications “pro”

It is not unusual when the life cycle of the major HENP applications last for dozens years.

In other words it is much more then the life cycle of any known “system components” like operating system, file system, GUI etc. and hardware as well.

To survive the sophisticated and expansive HENP applications must be in possession of some (tiny) layer to separate the HENP applications and the concrete system environment evolvement.

It is possible by introducing some sort of the “virtual operating system model” the applications rely on; and providing the implementation of that model as soon as operating environment evolves.

Operating system “model” for HENP cross-platform applications “cons”

Even though it does allow to prolong the life cycle of the software package that comes for the price.

- One is still required to learn and maintain the specific pieces of the code for many different platforms,
- and the different versions of one and the same platform for the entire application life.
- That sometimes entails the narrowing the number of different platforms the package is available for.

- On the other hand many features one has to deal with are not application specific.
- This implies that the good layer can be "borrowed" from some third party dedicated packag

Why Qt?

The present model has been serving ROOT and ROOT users very well for 9 years soon.

On the other hand, most tasks to be performed and implement have no ROOT specific and have been successfully solved by other packages.

- *Qt package from TrollTech AS* was especially attractive not only due to its superior quality and high level technical support but because it comes with the source code and tools to build it in place (including a commercial version for Windows).
- The rich set of Qt documentation can be found on Web and available from the leading publishers as well.
- Qt is a multi-platform C++ application framework that developers can use to write single-source applications that run-natively-on Windows, Linux, Unix, Mac OS X and embedded Linux.
- Qt has been used to build thousands of successful commercial applications worldwide,
- and is the basis of the open source KDE desktop environment.

Why Qt (cont.)

- Cross platforms
- Documentation
- Open Source (including Win32)
- The “native” development tool integration
- “Native” GUI (Win32, Mac) integration
- Free (for x11 and Mac)
- “non-commercial” for Win32
- Microsoft Foundation classes migration
- Motif migration
- Embedded platform

High quality

The review http://www.geocities.com/SiliconValley/Vista/7184/guitool.html of the more then 100 different GUI packages named Qt “the best GUI packages”

Qt – “softgate” to 3d party applications

- Qt – rich itself, but there is nothing up with that. All modern GUI systems (GDK, MFC, Motif etc) are in possession more/less the same functionality.
- Qt authors addressed many hot issues and the main one is how to connect the different packages within one application seamlessly?

Qt cornerstones

- Signal / Slot C++ object communication mechanism
- OO model: Qt uses the named generic container of the named generic containers. In other words Qt basic OO model is a the tree-like structure

<u>These allow significantly simplify the 3d-party interactive libraries integration.</u>

<u>They made it possible at last!</u>

Why now?
Qt and ROOT timeline

- **1994**: Trolltech founded in Oslo, Norway
- 1996: First sale of Qt (to European Space Agency)
- 1998: **Created the KDE Free Qt Foundation**
- 1998: Released Qt/X11 under the QPL
- 1999: Qt 2 released
- 1999: Opened Australian office
- 1999: Received first-round funding
- 2000: Released Qt/Embedded
- 2000: Opened US office
- 2000: Released Qt/X11 under the GPL
- 2001: First Qtopia adoption (by Sharp Electronics)
  2001: Qt 3 released
- **2002: Teambuilder released**
- 2003: QSA released
- 2003: Released Qt/Mac under the GPL
- 2003: Introduces Qt Solutions
- 2004: Released Qtopia Phone Edition

http://www.trolltech.com/company/history.html

- **Jan 95**: Thinking/writing/rewriting/???
- November 95: Public seminar, show Root 0.5
- Spring 96: decision to use CINT
- Jan 97: Root version 1.0
- **Jan 98: Root version 2.0 – STAR ROOT-based offline framework**
- Mar 99: Root version 2.21/08 (1st Root workshop FNAL)
- Feb 00: Root version 2.23/12 (2nd Root workshop CERN)
- Mar 01: Root version 3.00/06
- Jun 01: Root version 3.01/05 (3rd Root workshop FNAL)
- **Jan 02: Root version 3.02/07 (LCG project starts: RTAGs)**
- Oct 02: Root version 3.03/09 (4th Root workshop CERN)
- Dec 03: Root version 3.10/02 (last PRO release)
- Feb 04: Towards version 4.00 (5th Root workshop SLAC)
  http://www.slac.stanford.edu/BFROOT/www/Computing/Distributed/ROOT2004/files/brun.ppt

Approaching Qt

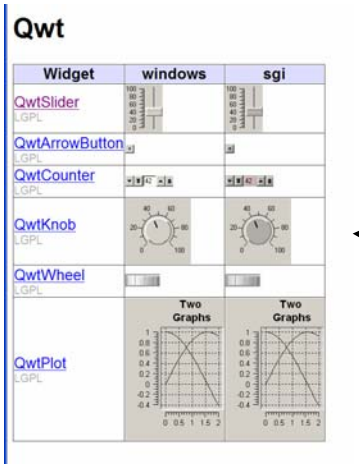
Goals:

- Immediate man-power conservation – “experiment target”
- Insure the package long healthy life (25 years?) – “developer target”

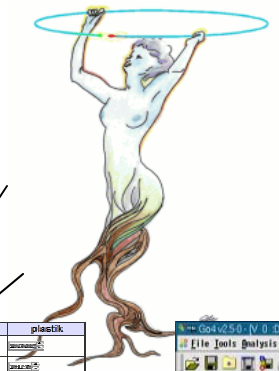
- Qt “alone” - LHC++ (CERN)
- Qt “outside” - GO4 approach (GSI)
- Qt “just inside” - Acat2002 (STAR)
- Qt “side by side” - Qt CINT approach (Masa Goto, HP Japan)
- Qt “deep inside” - Acat2003 (STAR)

Benefits

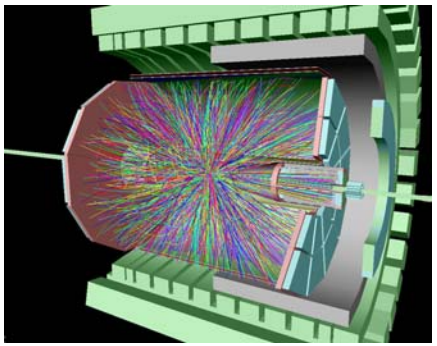
One can include Qt-based GUI components and packages into ROOT based applications



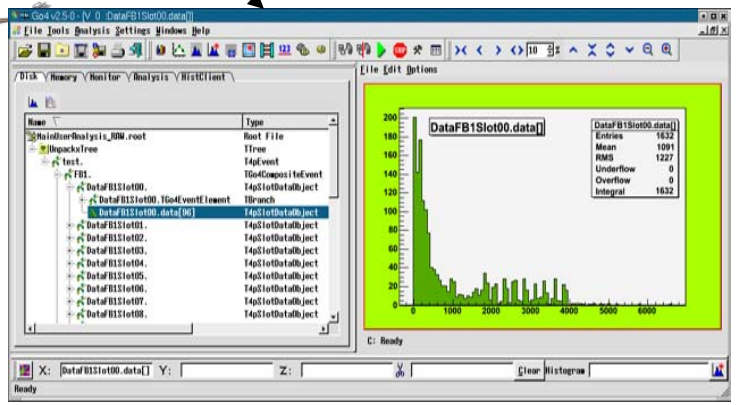
Qwt



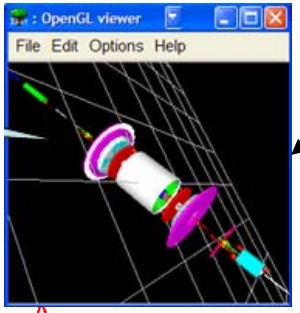
Coin3D

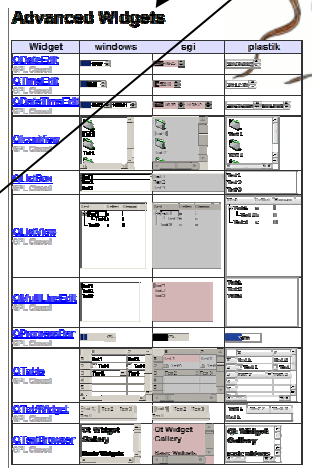


GO4

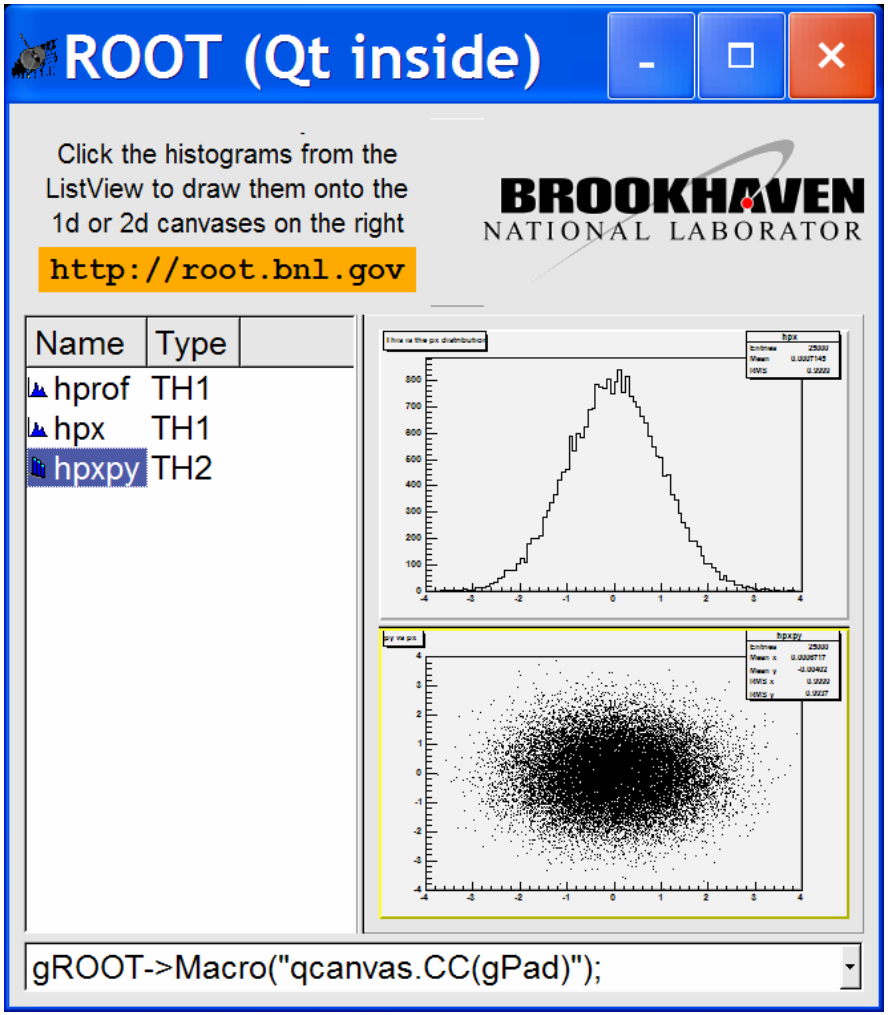


QGLViewer



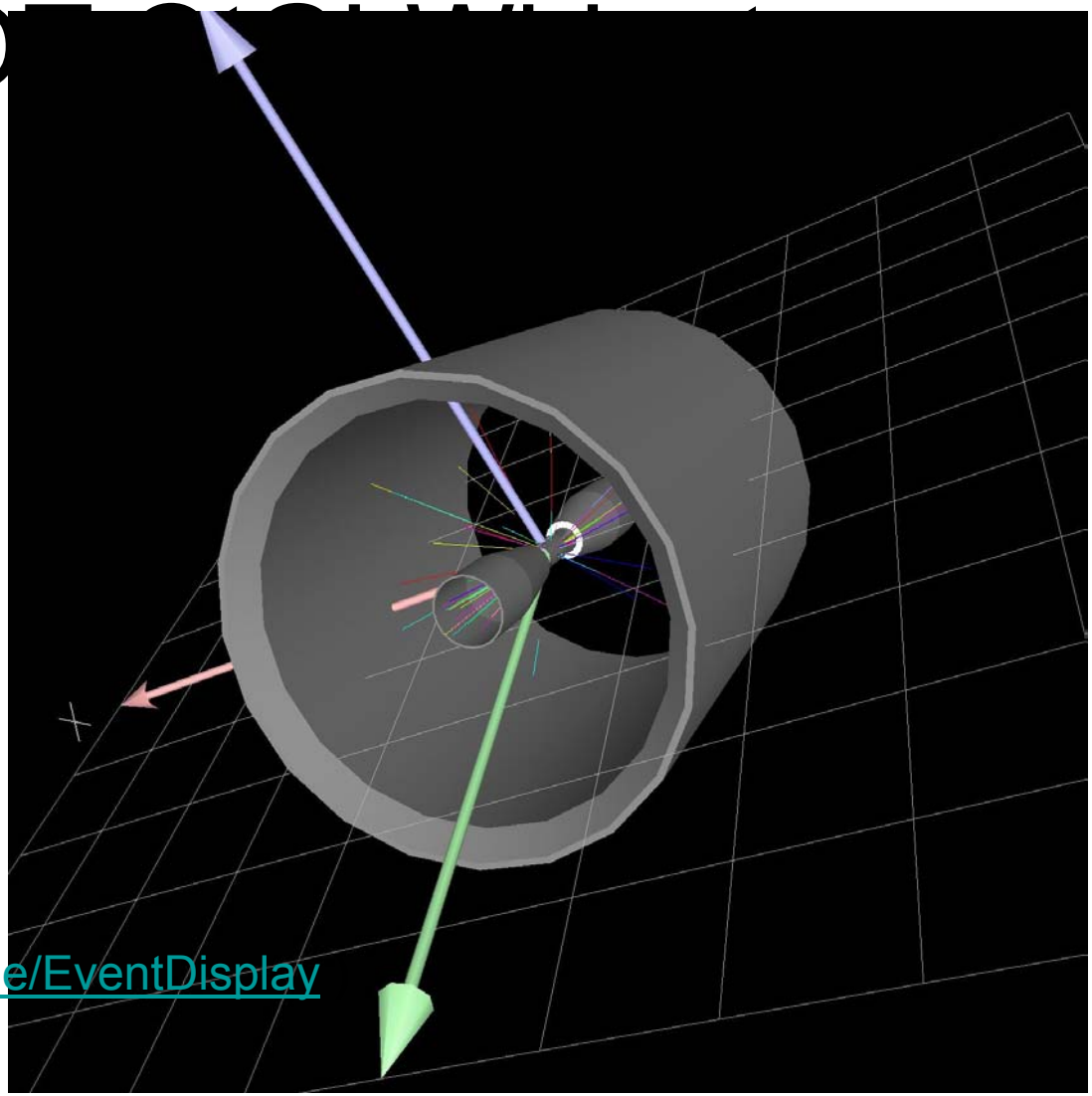


Simple histogram browser (60 lines of C++ code)



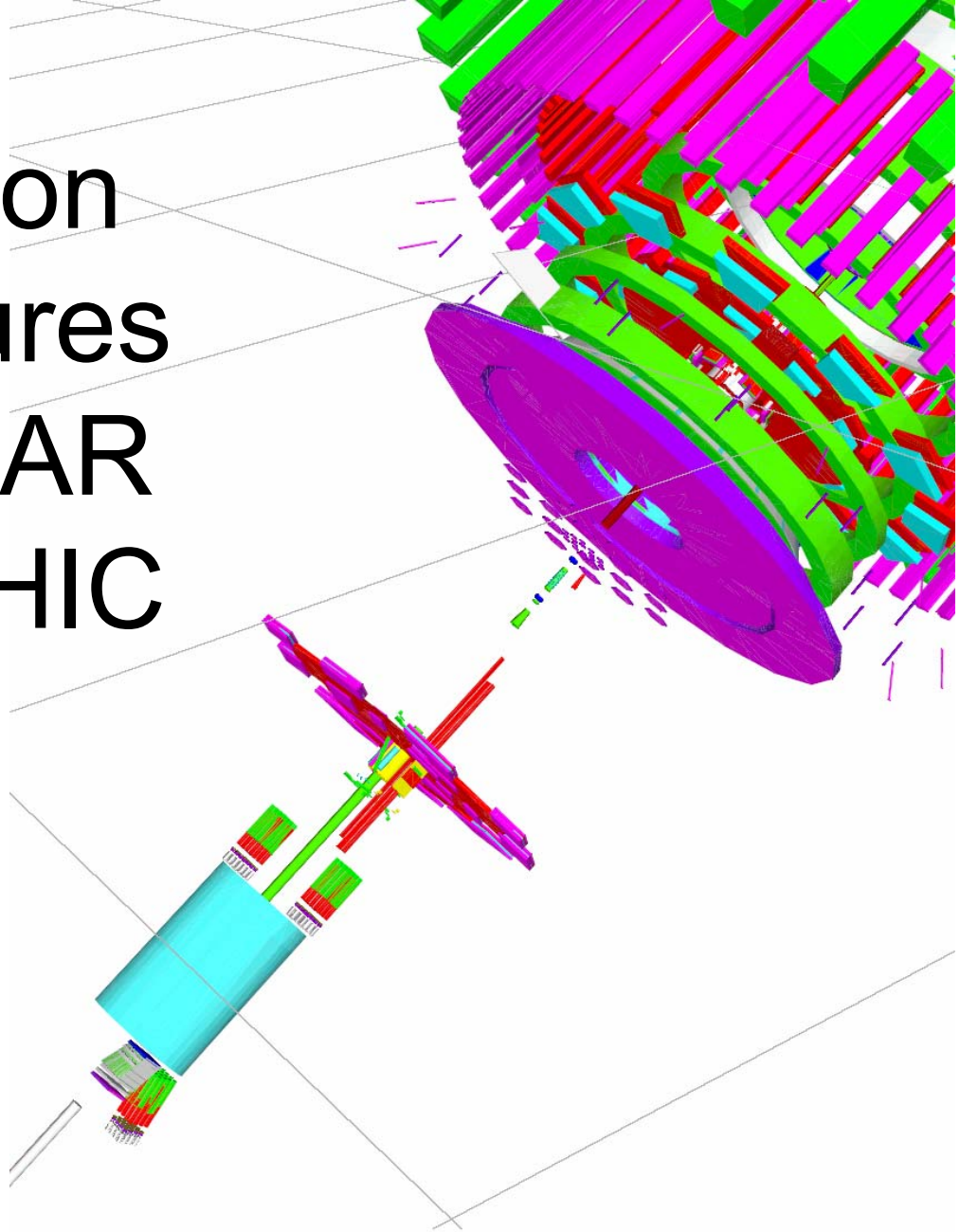
ROOT QtGLViewer

A fragment of STAR
“Event Display” QtGLViewer
class based viewer

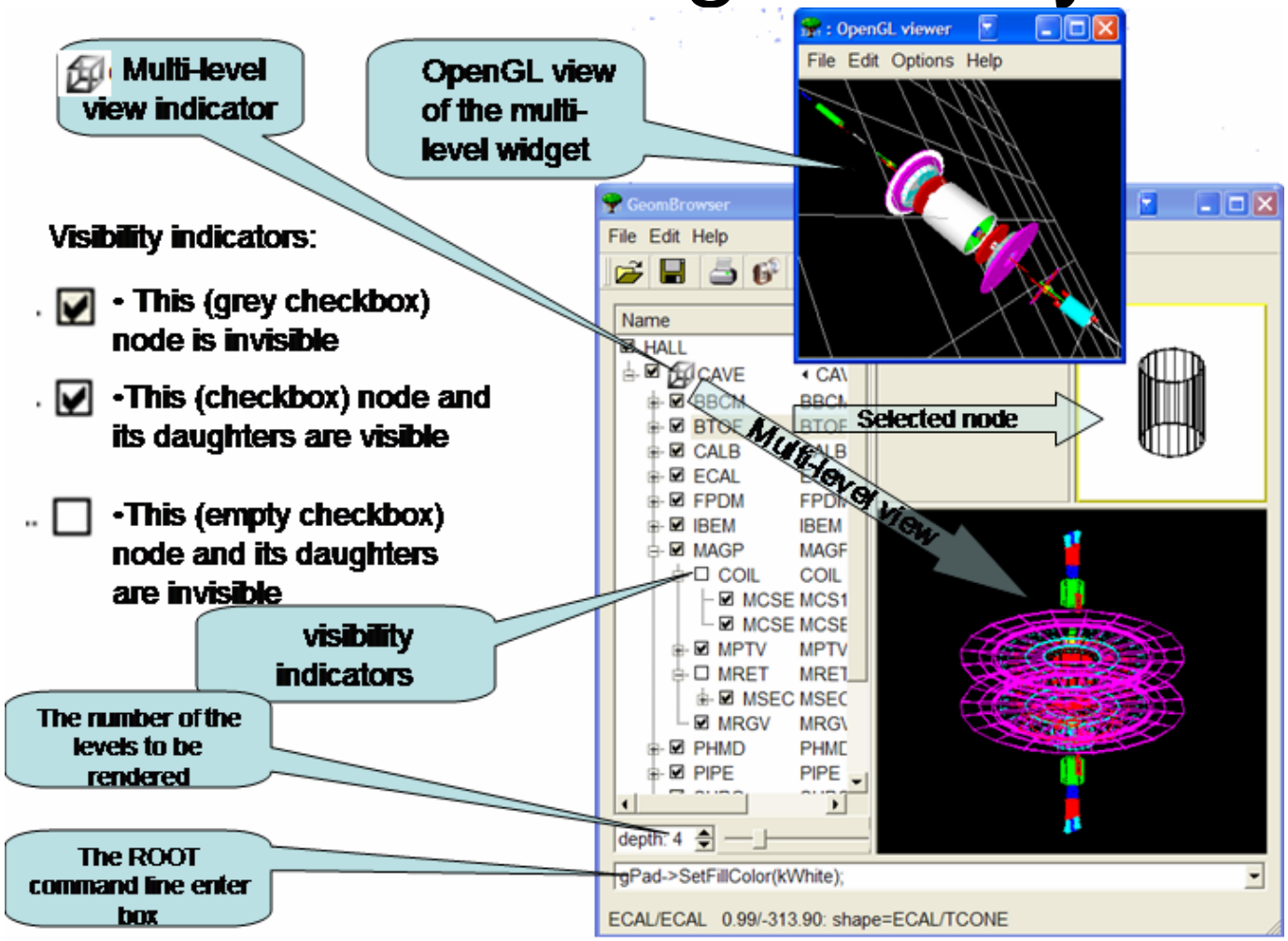


see: http://www.rhic.bnl.gov/~fine/EventDisplay

High resolution OpenGL pictures (fragment STAR detector at RHIC

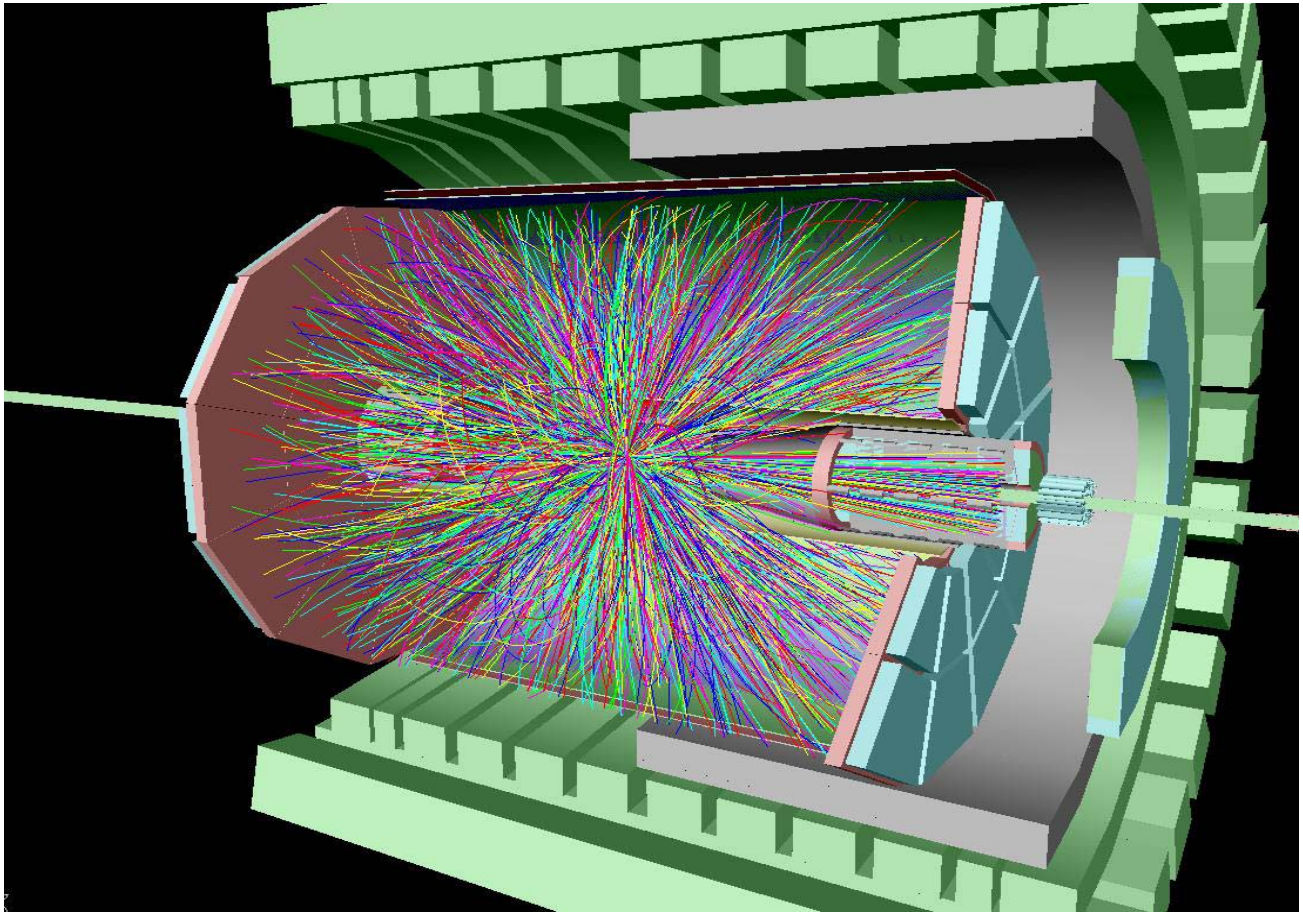


Qt-based ROOT geometry browser



STAR reconstructed events

It is also possible to use the Coin3D package to visualize ROOT objects



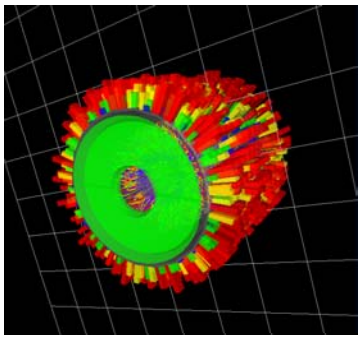
STAR Detector with real tracks

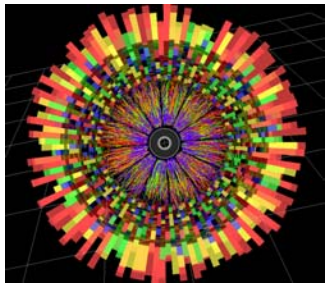
Offline EventDipslay

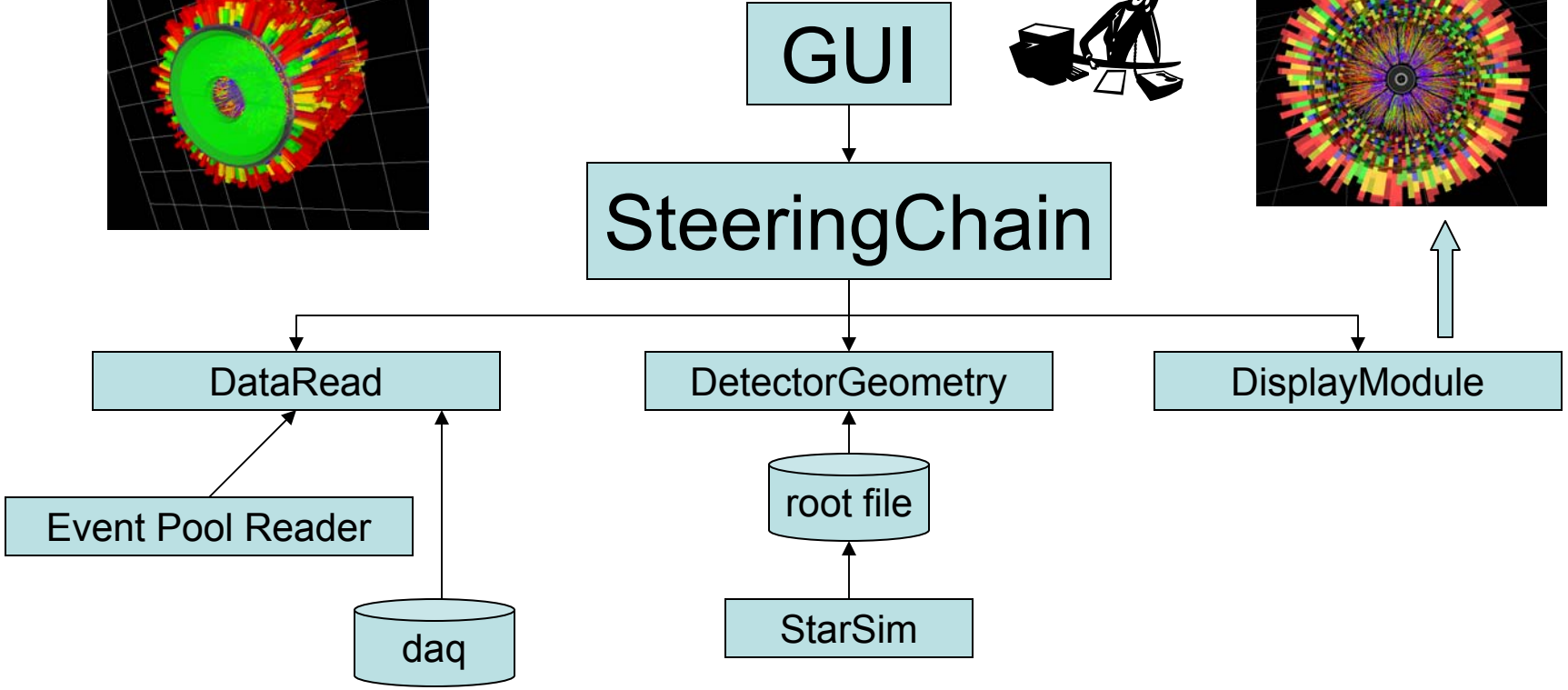
- Offline Event Display is a part of the STAR release for several releases.
- Populated with the new users' filters (developed by V.Perevoztchikov)
- "Calorimeter" hits representation developed for online display to be added soon.

Online EventDipslay

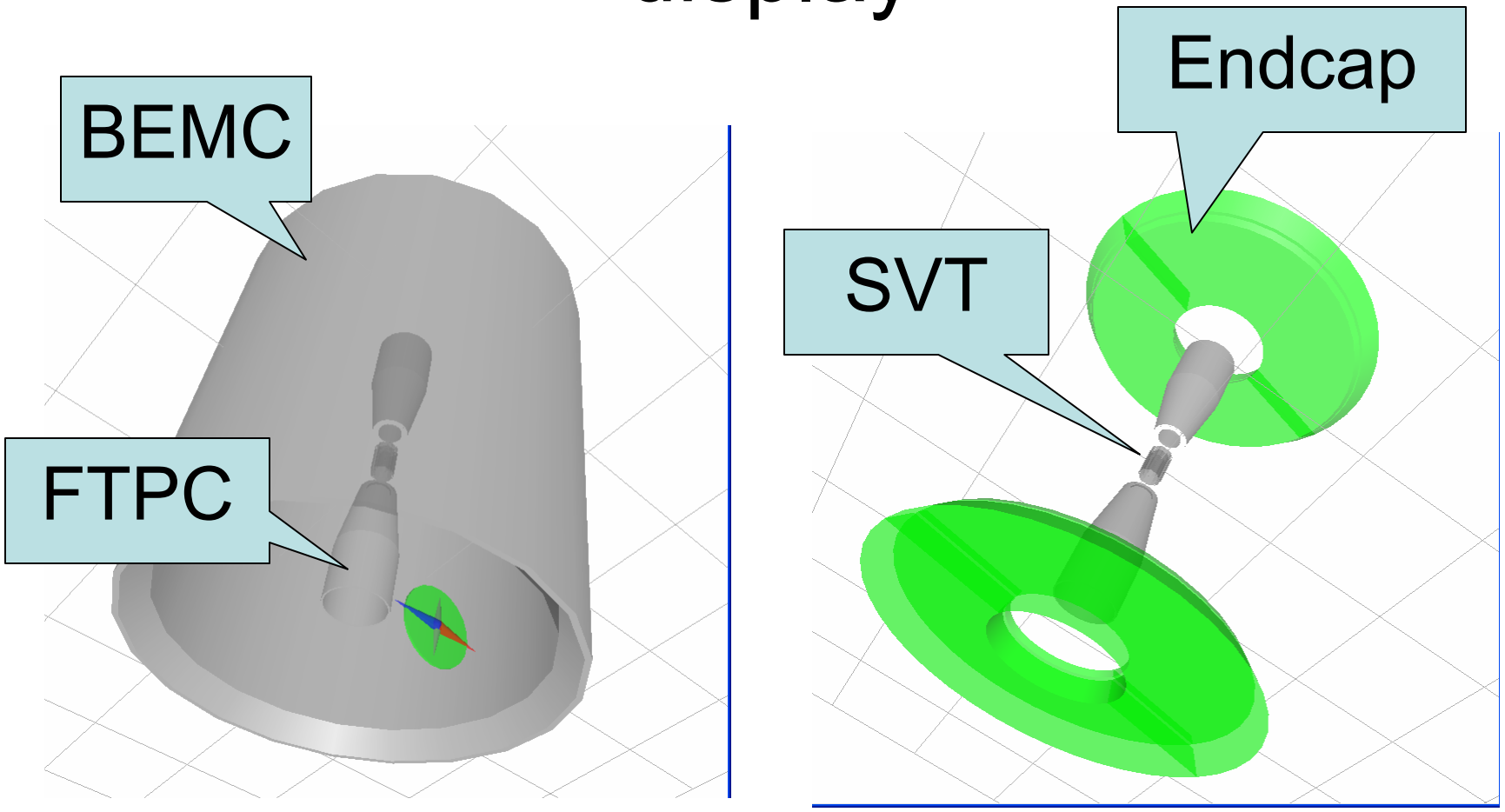
- Proven technology used for Offline Display



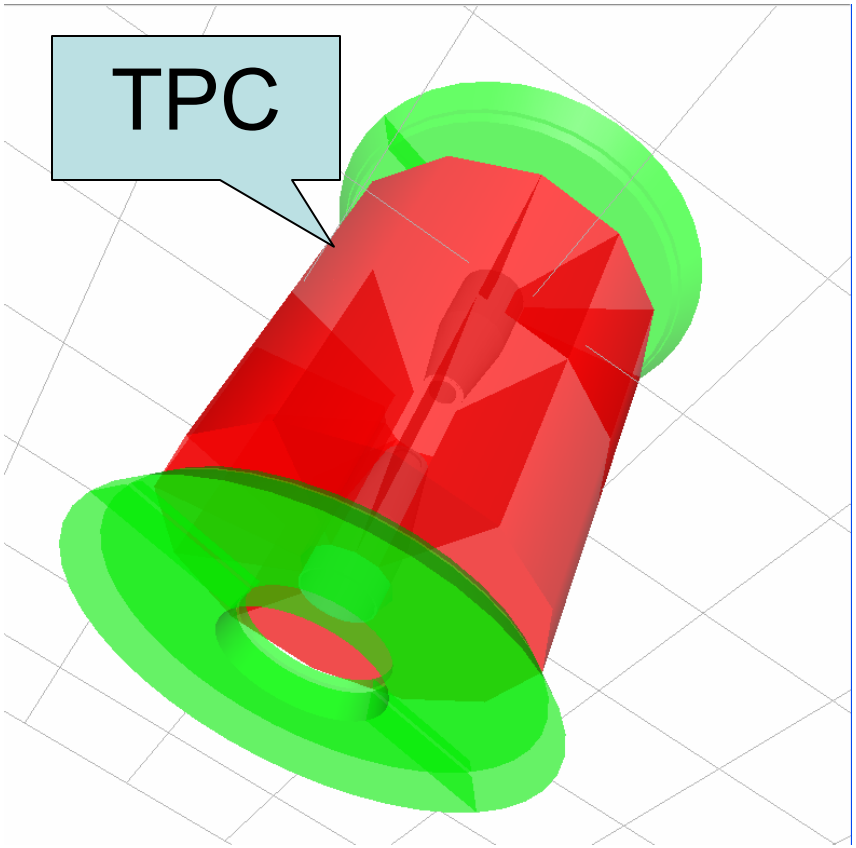


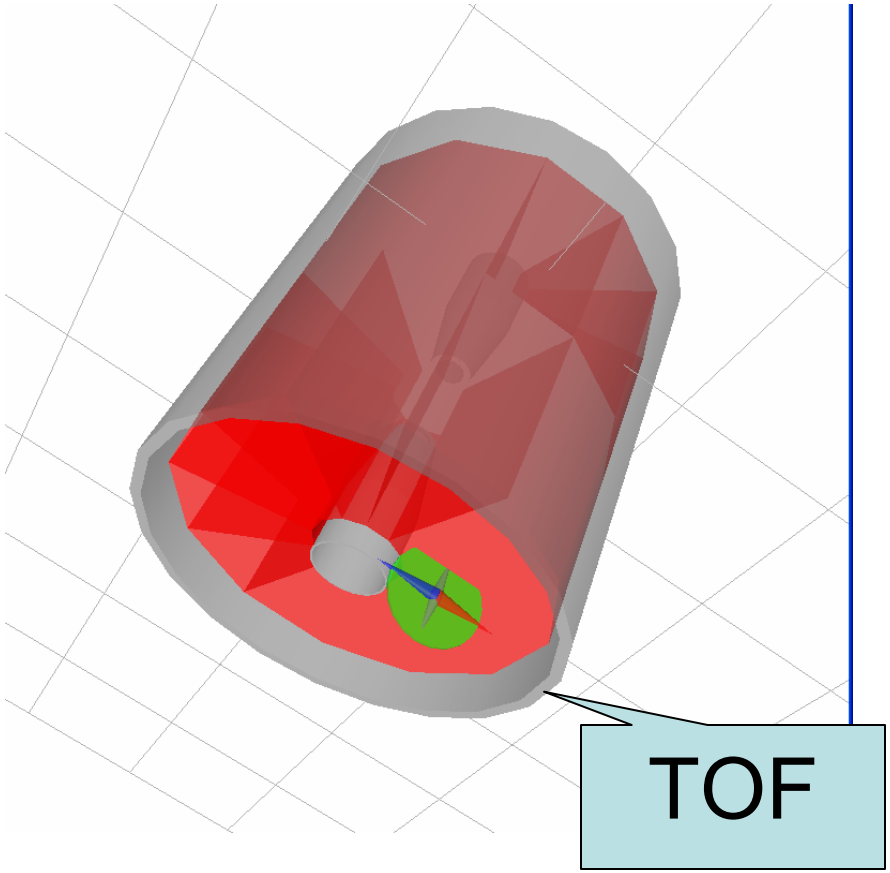


Geometry components of Online display



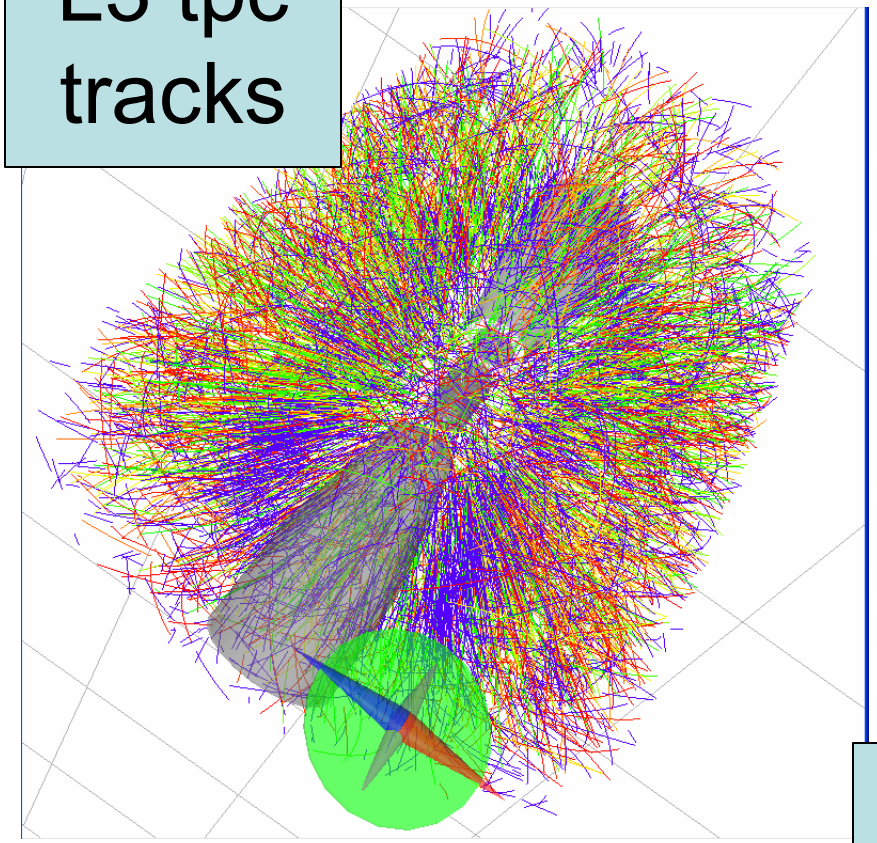
Geometry components of Online display

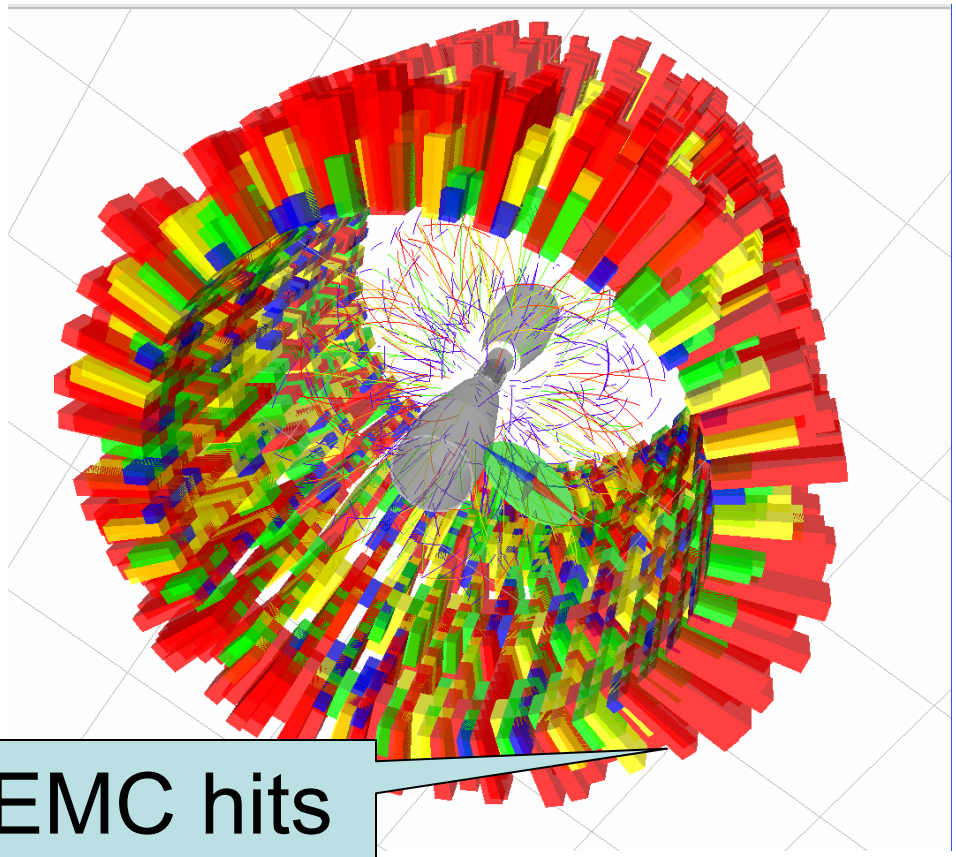




“Event components of Online display

L3 tpc tracks





EMC hits

Status

see: STAR -> Computing –> Infrastructure -> Visualization

- Qt (v 3.3.1) – standard component of the STAR env. Installed and available for all STAR official platforms.
- Qt-layer – included into the official ROOT (CERN) CVS repository CERN
- Qt-ROOT layer – included into STAR version of ROOT
- Offline display – part of the STAR releases
- Online display – installed on rts03.starp.bnl.gov node and available for “l3user” of that node.

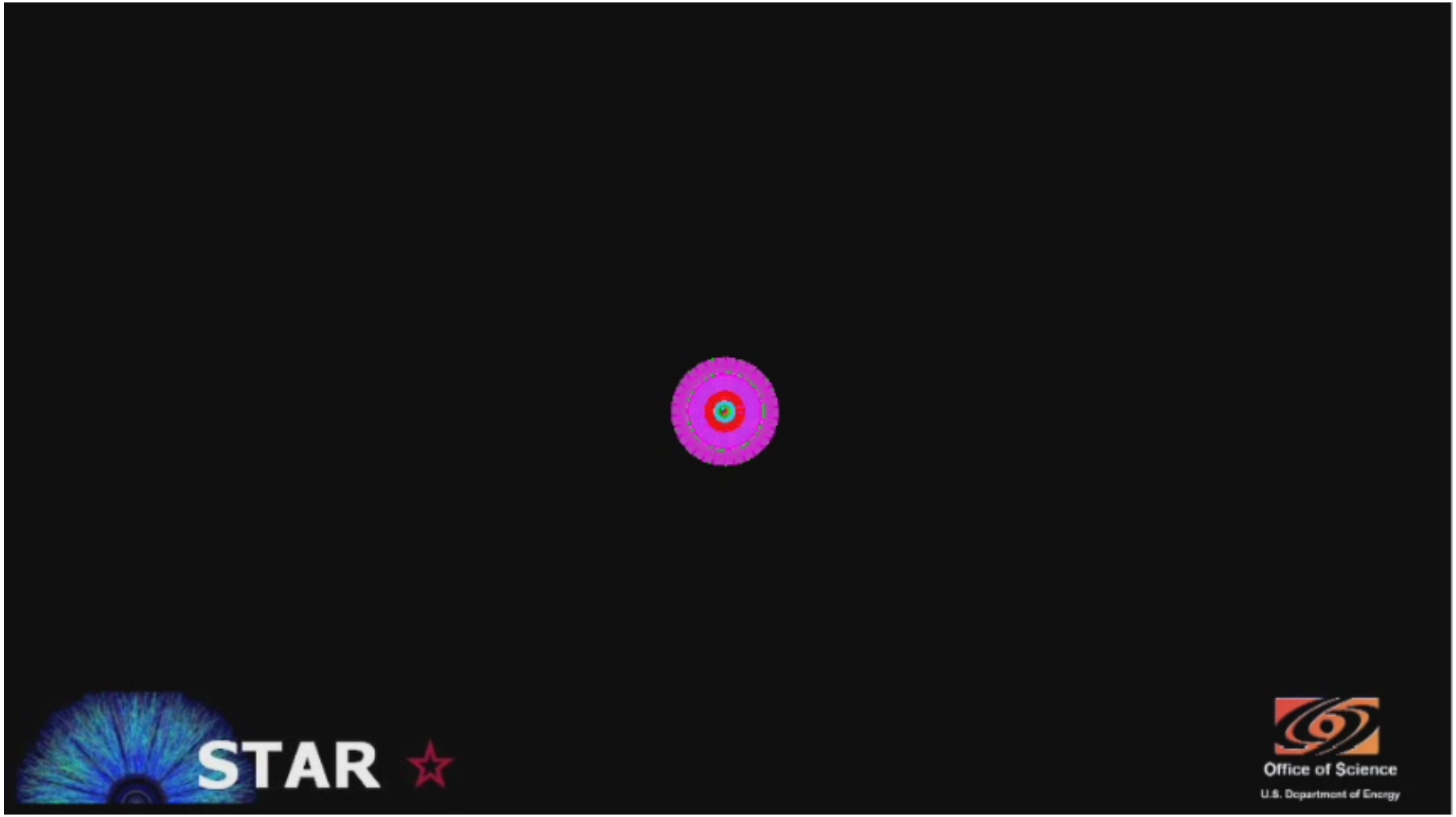
Conclusion

- At present STAR is in possession of the software infrastructure that does allow to create the “affordable” GUI application for Online, offline and data analysis applications.

Next

- Learn tools
- Contribute to the STAR “library” of the custom GUI components and stand-alone applications.

Au-Au Collision (seen at DOE booth)



References

- CINT status, Masa Goto
  http://www.slac.stanford.edu/BFROOT/www/Computing/Distributed/ROOT2004/files/goto.ppt
- Cross-platform approach to create the interactive application based on ROOT and Qt GUI libraries.
  http://www-conf.kek.jp/acat03/prog/presen/id0112.ppt
- ROOT in GO4, Joern Adamczewski
  http://www.slac.stanford.edu/BFROOT/www/Computing/Distributed/ROOT2004/files/adamczewski.ppt
- Cross-platform Qt-based implementation of low level GUI layer of ROOT
  http://acat02.sinp.msu.ru/presentations/fine/Acat2002.ppt
- Cross-platform approach to create the interactive application based on ROOT and Qt GUI libraries
  http://www-conf.kek.jp/acat03/prog/presen/id0112.ppt
- Visualization of the ROOT 3D class objects with Open Inventor-like viewers
  http://www-conf.kek.jp/acat03/prog/presen/id0113.ppt
- C++ GUI Programming with Qt3
  http://phptr.com/content/images/0131240722/downloads/blanchette_book.pdf